**Initial Idea Plan**

Exploration into Neural Networks

Making an AI learn to walk in a generation based rigid body simulation in Unity

* Multiple entities with pivot legs and arms, simulated weight and balance must try to learn to walk the furthest distance, storing parameters for each entity
* Each generation the best few are copied across to all
* An entity could fail if the top/head part touches the ground
* Exploration of types of neural networks, maybe comparing how two types learn at different rates
* Explore the effect of increased number of artificial neurons on the learning speed
* <https://towardsdatascience.com/building-a-neural-network-framework-in-c-16ef56ce1fef>

Making a chess simulator, then integrating a neural network-based Ai opponent

* Chess simulator would require multiple different types of pieces with different rules for each move
* Each piece could be done with separate child classes with each move as a function
* Neural network would need to know each possible move and the final goal, should make moves.
* Would need to know the effect of each of its moves and what it would lead to, maybe looking a few moves ahead.
* This would result in the Ai having to play against itself to check ahead

Working with an API

GPS route finding app using a map API

* Using a map API and a pathfinding algorithm to find routes to destinations
* Considering distance, number of turns and potentially exception e.g., motorways or traffic levels
* Could be worked into a mobile app, using unity

Making a train planner application, listing times, platforms, delays and routes between destinations using data from Darwin, a national information engine.

* Timings, lines and platforms would need to be collected
* Route planner would use a searching algorithm to find trains that stop at both designated platforms.
* Make a GUI for the app to visually display timings and allow the user to search for options
* Darwin API can be used to access the live data.

Game Making

Making a 2d procedurally generated Gameworld, in a terraria style

* Would use unity for the development
* Using purlin noise for terrain generation, in a code based mathematical generation
* Destructible world with crafting and building
* Game state/saves could be saved in a file
* Implement data structures for saving possessions and world alterations, e.g., list, stacks, dictionaries
* Add a complicated NPC and Enemy Ai
* Add online connection using port forwarding (Important for higher grades)
* Game optimisation

Making a physics simulator in c#